Liveness in L/U-Parametric Timed Automata

Étienne André and Didier Lime [AL17]

Université Paris 13, LIPN and École Centrale de Nantes, LS2N

Highlights, 14 September 2017, London, England
Introduction

- Parametric timed automata (PTA) allow for flexible, abstract, and robust modelling;
- The answer to parametric model-checking is appealing;
- Many undecidability results exist for safety / reachability properties;
- And a few decidable subclasses:
  - L/U PTA [HRSV02];
  - IP-PTA [ALR16];
  - bounded integer PTA [JLR15].
Introduction

- Parametric timed automata (PTA) allow for flexible, abstract, and robust modelling;
- The answer to parametric model-checking is appealing;
- Many undecidability results exist for safety / reachability properties;
- And a few decidable subclasses:
  - L/U PTA [HRSV02];
  - IP-PTA [ALR16];
  - bounded integer PTA [JLR15].
- What about liveness?
Parametric Timed Automata [AHV93]

\[
x = p_1 \\
a \\
x := 0
\]

\[
\begin{align*}
\ell_0 & \xrightarrow{a} \ell_0 \\
\ell_0 & \xrightarrow{x = 0 \land y \leq p_2, b} \ell_1 \\
y \leq p_2
\end{align*}
\]
Parametric Timed Automata [AHV93]

\[ x = p_1 \]
\[ a \]
\[ x := 0 \]

\[ x = 0 \land y \leq p_2, b \]
\[ y \leq p_2 \]

For \( p_1 = 1.2 \) and \( p_2 = 4 \):

- \( \ell_0 \):
  - \( x = 0 \)
  - \( y = 0 \)

- \( \ell_0 \) to \( \ell_0 \):
  - \( x = 1.2 \)
  - \( y = 1.2 \)

- \( \ell_0 \) to \( \ell_1 \):
  - \( x = 0 \)
  - \( y = 1.2 \)

- \( \ell_1 \):
  - \( x = 0 \)
  - \( y = 1.2 \)

- \( \ell_1 \) to \( \ell_1 \):
  - \( x = 2.4 \)
  - \( y = 3.6 \)
L/U Parametric Timed Automata [HRSV02]

- Parameters are used either as **lower** bounds or as **upper** bounds, never both.
- **Monotonicity**: increasing upper bounds or decreasing lower bounds gives **more** behaviours.
Liveness in (Parametric) Timed Automata

- Our **liveness** properties concern **maximal** paths:
  - Existence of an **infinite** maximal path (discrete **cycle**, denoted EC);
  - Existence of a **finite** maximal path (**deadlock**, denoted ED);
  - Existence of a maximal path preserving some property (**CTL EG** property).
Liveness in (Parametric) Timed Automata

▶ Our liveness properties concern maximal paths:
  ▶ Existence of an infinite maximal path (discrete cycle, denoted EC);
  ▶ Existence of a finite maximal path (deadlock, denoted ED);
  ▶ Existence of a maximal path preserving some property (CTL $\text{EG}$ property).

▶ Parametric properties:
  ▶ $\phi$-emptiness: is the set of parameter valuations s.t. $\phi$ holds empty?
  ▶ $\phi$-universality: is the set of parameter valuations s.t. $\phi$ holds universal?
Liveness in (Parametric) Timed Automata

- Our **liveness** properties concern maximal paths:
  - Existence of an **infinite** maximal path (discrete **cycle**, denoted EC);
  - Existence of a **finite** maximal path (**deadlock**, denoted ED);
  - Existence of a maximal path preserving some property (CTL **EG** property).

- **Parametric** properties:
  - $\phi$-emptiness: is the set of parameter valuations s.t. $\phi$ holds **empty**?
  - $\phi$-universality: is the set of parameter valuations s.t. $\phi$ holds universal?
## Results from the Literature

<table>
<thead>
<tr>
<th>Class</th>
<th>PTA</th>
<th>L/U PTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC-emptiness</td>
<td>open</td>
<td><strong>PSPACE-c.</strong></td>
</tr>
<tr>
<td>ED-emptiness</td>
<td>open</td>
<td>open</td>
</tr>
<tr>
<td>EG-emptiness</td>
<td>open</td>
<td>open</td>
</tr>
</tbody>
</table>

\[1\text{Integer parameters [BL09].}\]
EC-emptiness is PSPACE-c for L/U PTAs

- There exists a **rational** parameter valuation s.t. there is a cycle iff there exists an **integer** valuation.
- Use the **monotonicity** property of L/U PTAs: **round** up for upper bounds, down for lower bounds to get a good **integer** valuation.
EC-emptiness is *undecidable* for PTAs

- Reduce from the *counter boundedness* problem of 2-counter machines
  - Finite-state machine + 2 non-negative integer counters;
  - *increment* some counter and go to some state;
  - *if* some counter is *zero* then *decrement* it and go to some state; otherwise go to some other state;
  - Halting: $q_{halt}$

There is a (discrete) cycle in the PTA iff the counters are bounded:
- if the machine halts, $q_{halt}$ is reachable $\rightarrow$ cycle;
- if the machine does not halt but the counters are bounded, there is a parameter valuation small enough to have a cycle among the instruction widgets;
- if the counters are unbounded, for any valuation, the PTA will eventually block in the increment widget.
EC-emptiness is **undecidable** for PTAs

- Reduce from the **counter boundedness** problem of 2-counter machines
  - Finite-state machine + 2 non-negative integer counters;
  - increment some counter and go to some state;
  - if some counter is zero then decrement it and go to some state; otherwise go to some other state;

- Halting:

  ![Diagram](image)
EC-emptiness is *undecidable* for PTAs

- Reduce from the *counter boundedness* problem of 2-counter machines
  - Finite-state machine + 2 non-negative integer counters;
  - *increment* some counter and go to some state;
  - if some counter is zero then *decrement* it and go to some state; otherwise go to some other state;

- Halting:

  ![Diagram](image)

  - There is a (discrete) cycle in the PTA iff the counter are bounded:
    - if the machine halts, $q_{halt}$ is reachable $\rightarrow$ cycle;
    - if the machine does not halt but the counters are bounded, there is a parameter valuation *small enough* to have a cycle among the instruction widgets;
    - if the counters are unbounded, for any valuation, the PTA will eventually *block* in the increment widget.
ED-emptiness is *undecidable* for L/U PTAs

- Reduce from the *halting* problem of 2-counter machines;
ED-emptiness is *undecidable* for L/U PTAs

- Reduce from the **halting** problem of 2-counter machines;
- Change previous construction to “split” parameters and get an L/U PTA
  - Replace $x \leq p$ with $x \leq p^+$
  - Replace $x \geq p$ with $x \geq p^-$
  - Replace $x = p$ with $p^- \leq x \leq p^+$
- We use the deadlock property to enforce $p^- = p^+$. 
EG-emptiness is *undecidable* for L/U PTAs

- by reduction from the *halting* problem of 2-counter machines;
- similar to the ED-construction with a different encoding adapted from [BBLS15];
- the main idea is to eliminate cycles by:
  - making sure all widgets execute in 1 t.u.;
  - add a global invariant limiting the *total execution time* so that it does not exceed some parameter $p_2$;
  - then the PTA can only execute *at most* $p_2$ instructions and $p_2$ has to be *big enough* for executing a halting sequence.
### Bounded parameters

#### Results up to now

<table>
<thead>
<tr>
<th>Class</th>
<th>PTA</th>
<th>L/U PTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC-emptiness</td>
<td>Undec.</td>
<td>PSPACE-c.</td>
</tr>
<tr>
<td>ED-emptiness</td>
<td>Undec.</td>
<td>Undec.</td>
</tr>
<tr>
<td>EG-emptiness</td>
<td>Undec.</td>
<td>Undec.</td>
</tr>
</tbody>
</table>

We can find some decidability by considering parameters are bounded (each takes its values in some bounded interval); changes nothing for PTA.

We consider both (topologically) closed and open parameter domains.
Results up to now

We can find some decidability by considering parameters are **bounded** (each takes its values in some bounded interval);

- Changes nothing for **PTAs**;
- We consider both (topologically) **closed** and **open** parameter domains.
EG-emptiness is **decidable** for closed bounded L/U PTA

1. Test if there is an infinite path preserving $\phi$ in the TA obtained by setting:
   - **lower** bounds to their **minimum** value,
   - and upper bounds to their maximal values.
   i.e. verify CTL property “EG ($\phi \land \text{EX true}$)” on the **region graph** of the TA.

2. if yes we are done
EG-emptiness is **decidable** for closed bounded L/U PTA

1. Test if there is an infinite path preserving $\phi$ in the TA obtained by setting:
   - **lower** bounds to their **minimum** value,
   - and upper bounds to their maximal values.
   i.e. verify CTL property “$\text{EG} \ (\phi \land \text{EX true})$” on the **region graph** of the TA.
2. if yes we are done
3. otherwise all paths preserving $\phi$ are finite: explore them symbolically, using the **symbolic polyhedral abstraction** of linear hybrid automata;
4. test all symbolic states on those paths for **deadlocks**:
   - consider all states that can reach some guard (classic past operator)
   - check if those states **cover** the whole symbolic state (polyhedral union and inclusion).
EG-emptiness is \textit{undecidable} for open bounded L/U PTA

- Reduce from the \textbf{halting} problem of \textit{2-counter machines}
- Make sure all widgets execute in $[p_2^-, p_2^+]$ t.u. (instead of 1);
- use the \textbf{open} parameter domain to enforce $p_2^- > 0$;
- add a \textbf{global invariant} so that the whole PTA can only execute for 1 t.u. to \textbf{eliminate cycles};
- the machine \textbf{halts} iff there exists a parameter valuation s.t. $p_1^- = p_1^+$ and $p_2^- = p_2^+$ and there is a \textbf{deadlock} in the PTA.
## Final Results

The other results follow directly from the previous constructions;

- We conjecture that EC-emptiness for open bounded L/U PTAs is **decidable** with techniques similar to [San11].

<table>
<thead>
<tr>
<th>Class</th>
<th>PTA</th>
<th>L/U PTA</th>
<th>closed b. L/U</th>
<th>open b. L/U</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC-empt.</td>
<td>Undec.</td>
<td>PSPACE-c.</td>
<td>PSPACE-c.</td>
<td>open</td>
</tr>
</tbody>
</table>
Conclusion and Perspectives

▶ Summary:
  ▶ We have exhibited a very thin border of decidability for liveness properties;
  ▶ It depends on the boundedness of the parameters and the topological closure of their initial domain.

▶ Future work:
  ▶ Prove that EC-emptiness for open bounded LU PTAs is decidable;
  ▶ Complete the results for the universality problems;
  ▶ Find the complexity of EG-emptiness for closed bounded L/U PTA.
References I

Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi.
Parametric real-time reasoning.

Étienne André and Didier Lime.
Liveness in L/U-parametric timed automata.
To appear.

Étienne André, Didier Lime, and Olivier H. Roux.
Decision problems for parametric timed automata.

Nikola Beneš, Peter Bezděk, Kim G. Larsen, and Jiří Srba.
Language emptiness of continuous-time parametric timed automata.


EC-emptiness is PSPACE-c for L/U PTAs

- There exists a rational parameter valuation s.t. there is a cycle iff there exists an integer valuation.
- Use the monotonicity property of L/U PTAs: round up for upper bounds, down for lower bounds to get a good integer valuation.
EC-emptiness is *undecidable* for PTAs

- Reduce from the **counter boundedness** problem of *2-counter machines*
  - Finite-state machine + 2 non-negative integer counters;
  - **increment** some counter and go to some state;
  - **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;
EC-emptiness is *undecidable* for PTAs

- Reduce from the **counter boundedness** problem of 2-counter machines
  - Finite-state machine + 2 non-negative integer counters;
  - *increment* some counter and go to some state;
  - *if* some counter is zero then *decrement* it and go to some state; otherwise go to some other state;
- States of the machines are encoded by locations $q_i$;
- Counters are encoded by clocks $y, z$ and one parameter $p$: when clock $x$ is null,
  \[
  \begin{align*}
  y &= 1 - c_1 p \\
  z &= 1 - c_2 p
  \end{align*}
  \]
EC-emptiness is **undecidable** for PTAs

- Reduce from the **counter boundedness** problem of 2-counter machines
  - Finite-state machine + 2 non-negative integer counters;
  - **increment** some counter and go to some state;
  - **if** some counter is zero then **decrement** it and go to some state; otherwise go to some other state;
- States of the machines are encoded by locations $q_i$;
- Counters are encoded by clocks $y, z$ and one parameter $p$: when clock $x$ is null,
  $$y = 1 - c_1 p$$
  $$z = 1 - c_2 p$$
- Initialisation:
  $$x = p \land x > 0$$
  $$x = 1$$
  $$x := 0$$
Details on the encoding for EC-emptiness

EC-emptiness is *undecidable* for PTAs

- Increment:

```
z = 1
z := 0
```

```
x = 0
```

```
y = p + 1
y := 0
x = 1
x := 0
```

```
q_i \rightarrow l_{i1}
```

```
\rightarrow l_{i2}
```

```
y = p + 1
y := 0
```

```
z = 1
z := 0
```

```
\rightarrow l'_{i2}
```

```
\rightarrow l_{i3}
```

```
\rightarrow q_j
```

\[\text{implies } p \leq 1 \text{ or it blocks at } l_{i3}.\]
EC-emptiness is *undecidable* for PTAs

- Increment:

\[
\begin{align*}
q_i &\rightarrow l_{i1} & l_{i1} &\rightarrow l_{i2} & l_{i2} &\rightarrow l_{i3} & l_{i3} &\rightarrow q_j \\
\end{align*}
\]

\[
\begin{align*}
\text{if } x = 0 &\text{ then } y = p + 1 \\
\text{if } y = 0 &\text{ then } z = 1 \text{ (and } x) \\
\text{if } z = 0 &\text{ then } y = 1 \text{ (and } x) \\
\end{align*}
\]

Details on the encoding for EC-emptiness
EC-emptiness is **undecidable** for PTAs

- Increment:

  - $q_i \rightarrow l_{i1}$
    - $x = 0$
    - $y = 1 - c_1 p$
    - $z = 1 - c_2 p$
    - $l_{i1} \rightarrow 0 l_{i2}$
      - $x = 0$
      - $y = 1 - c_1 p$
      - $z = 1 - c_2 p$
    - $l_{i2} \rightarrow c_2 p l_{i3}$
      - $x = c_2 p$
      - $y = 1 - (c_1 - c_2) p$
      - $z = 0$
    - $l_{i3} \rightarrow (c_1 + 1) p q_j$
      - $x = (c_1 + 1) p$
      - $y = 0$
      - $z = (c_1 - c_2 + 1) p$
  - $l_{i2} \rightarrow \rightarrow l'_{i2}$
    - $y = p + 1$
    - $y := 0$
    - $z = 1$
    - $z := 0$
  - $l'_{i2} \rightarrow \rightarrow l_{i3}$
    - $x = 0$
    - $y = 1 - (c_1 + 1) p$
    - $z = 1 - c_2 p$
    - $l_{i3} \rightarrow \rightarrow q_j$
      - $x = 0$
      - $y = 1 - (c_1 + 1) p$
      - $z = 1 - c_2 p$

- **implies** $p \leq \frac{1}{c_1 + 1}$ otherwise it **blocks** at $l_{i3}$.
EC-emptiness is *undecidable* for PTAs

- **Zero-test and decrement:**

  ![Diagram](image)

  - $c_1 = 0$ iff $y = 1$.
  - Decrement is similar to increment.
EC-emptiness is *undecidable* for PTAs

- Halting:
  - If the machine halts, $q_{\text{halt}}$ is reachable → cycle;
  - If the machine does not halt but the counters are bounded, there is a parameter valuation small enough to have a cycle among the instruction widgets;
  - If the counters are unbounded, for any valuation, the PTA will eventually block in the increment widget.
Details on the encoding for EC-emptiness

EC-emptiness is *undecidable* for PTAs

- **Halting:**

- There is a (discrete) cycle in the PTA iff the counter are bounded:
  - if the machine halts, $q_{halt}$ is reachable $\rightarrow$ cycle;
  - if the machine does not halt but the counters are bounded, there is a parameter valuation *small enough* to have a cycle among the instruction widgets;
  - if the counters are unbounded, for any valuation, the PTA will eventually *block* in the increment widget.
ED-emptiness is *undecidable* for L/U PTAs

- Reduce from the **halting** problem of 2-counter machines;
ED-emptiness is **undecidable** for L/U PTAs

- Reduce from the **halting** problem of 2-counter machines;
- Change previous construction to “split” parameters and get an L/U PTA:

  - We use the deadlock property to **enforce** $p^- = p^+$.
EG-emptiness is *undecidable* for open bounded L/U PTA

- Reduce from the **halting** problem of 2-counter machines
- Make sure all widgets execute in \([p^-_2, p^+_2]\) t.u. (instead of 1);

\[
\begin{align*}
p^-_2 &\leq z \leq p^+_2 \\
z &:= 0
\end{align*}
\]

\[
\begin{align*}
p^+_1 + p^-_2 &\leq y \\
y &:= 0
\end{align*}
\]

- use the **open** parameter domain to enforce \(p^-_2 > 0\);
- add a **global invariant** so that the whole PTA can only execute for 1 t.u. to eliminate cycles;
- the machine **halts** iff there exists a parameter valuation s.t. \(p^-_1 = p^+_1\) and \(p^-_2 = p^+_2\) and there is a **deadlock** in the PTA.